REMARKS

Applicant is in receipt of the Office Action mailed May 28, 2004. Reconsideration of the present case is earnestly requested in light of the following remarks.

The Office Action states: "Applicant's preliminary amendment adding claims 36-66 have been received and entered on 23 July 2002. However, it is noted that on page 7 of the preliminary amendment, applicant states that claims 36-65 have been added. . .since claims 36-66 have been added in actuality, it would be appreciated if the applicant makes the corresponding changes."

Applicant acknowledges Examiner's note and makes the corresponding changes based on the following. Applicant respectfully submits that, in the Preliminary Amendment mailed on 18 July 2002 and entered on 23 July 2002, claims 36-66 were added to more fully and completely claim embodiments of Applicant's invention. Accordingly, Applicant respectfully submits that claims 1-66 are pending in the Application.

Objection to the Abstract

The Abstract was objected to as not being in the proper form. Applicant has amended the Abstract to be within the range of 50 to 150 words. Applicant respectfully submits that the Abstract, as amended, is in proper form.

§102 Rejections

Claims 1-7 and 10-66 were rejected under 35 U.S.C. 102(b) as being anticipated by Kodosky et al. (U.S. Pat. No. 5,301,336, hereinafter "Kodosky"). This rejection is respectfully traversed.

As the Examiner is certainly aware, anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The identical invention must be

shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

Applicant respectfully submits that Kodosky nowhere teaches or suggests ". . .wherein the graphical source code is operable to execute in response to a user interface event. . ." as currently recited by Applicant's claim 36.

Rather, Kodosky teaches and discloses polling for external events such as a passage of time or a signal from an I/O channel: "Code virtual instruments which poll for an external event (passage of time, signal from a I/O channel etc) are implemented such that each atomic execution of an increment of code checks for the polled condition once on behalf of all nodes queued to the virtual instrument" (Kodosky col. 35, line 68 - col. 36, line 4) (*emphasis added*).

In contrast, Applicant's invention as recited in claim 36 includes ". . .wherein the graphical source code is operable to execute in response to a user interface event. . ." (*emphasis added*). Kodosky nowhere teaches or suggests this feature.

As clearly described in the background section of the present Application (page 5, line 9), and as is well known to those skilled in the art of programming, "event-driven" or "event-based" programming is a term of art, and refers to particular programming models and methodologies wherein program flow is based on "events", and is in direct contrast with the traditional "sequential" programming model. This event-driven programming model is exemplified by many applications that run in or under windowed operating systems such as Apple Computer's MacOS and Microsoft Corporation's Windows operating systems supporting window-based user interaction with the applications, where, for example, the programs respond to user interface events such as mouse and keyboard events, among others. Thus, in the context of the present Application the term "user interface event" is a specific term of art.

Thus, Applicant respectfully submits that claim 36 is patentably distinguished over Kodosky. Accordingly, Applicant respectfully submits that, at least for one or more reasons presented, claim 36 and those dependent therefrom are allowable.

Claims 53 and 66 include limitations similar to claim 36, and so the arguments presented above apply with equal force to these claims, as well. Applicant respectfully submits that, for at least one or more reasons presented, claims 53 and 66, and any claims, respectively, dependent thereon are patentably distinguished over Kodosky and are allowable.

The Office Action asserts: ". . .Kodosky et al. teach. . .displaying an event structure node in a block diagram for the graphical program in response to user input (user designing and interacting with the block diagram, which is made up of connected nodes, or icons, as shown in Figure 22). . ." (*emphasis added*). Applicant respectfully disagrees. Applicant respectfully submits that Kodosky does not disclose an "event structure node".

The Office Action generally refers to FIG. 22 of Kodosky as teaching the "event structure node" in a block diagram. Applicant is unsure as to what specific element or node in FIG. 22 of Kodosky is considered to teach the "event structure node". Moreover, Applicant respectfully submits that Kodosky's col. 17, line 48 - col. 18, line 32 describe features of Kodosky's FIG. 22 and do not teach or suggest an "event structure node". Applicant respectfully submits that Kodosky's Figure 22 is simply an illustration of an example graphical program that contains no event structure node.

In contradistinction, Applicant's invention as recited in claim 1 includes ". . . displaying an event structure node in a block diagram for the graphical program in response to user input. . ." Kodosky nowhere teaches or suggests this feature. Further, Applicant notes that nowhere does Kodosky teach or suggest an "event structure node" as disclosed in the present application, where the event structure node is configurable to "receive and respond to one or more user interface events during execution of the graphical program".

Furthermore, Applicant respectfully submits that Kodosky nowhere teaches ". . .creating a graphical user interface for the graphical program in response to user input. . .(*emphasis added*)" as recited in Applicant's claim 1.

Rather than creating a graphical user interface for a graphical program, Kodosky teaches a graphical program sequentially executes an ordered and pre-determined set of tasks, for performing some action. In other words, Kodosky teaches executing a graphical program (i.e. block diagram) to execute an ordered set of tasks for performing some action or a procedure:

> A method for programming a computer to execute a procedure, is based on a graphical interface which utilizes data flow diagrams to represent the procedure. The method stores a plurality of executable functions, scheduling functions, and data types. A data flow diagram is assembled in response to the user input utilizing icons which correspond to the respective executable functions, scheduling functions, and data types. . .An executable program is generated in response to the data flow diagram and the panel utilizing the executable functions, scheduling functions, and data types stored in the memory. (Kodosky Abstract) (*emphasis added*).

The Office Action suggests that Kodosky's col. 17, lines 48-54 teach ". . .creating a graphical user interface for the graphical program in response to user input. . .(*emphasis added*)" Rather, Applicant respectfully submits that Kodosky's col. 17, lines 48-54 teach a graphical program is displayed in a graphical integrated development environment (IDE) where the graphical IDE includes a graphical user interface.

Applicant respectfully submits that the graphical IDE or graphical user interface is not the graphical program. This can be appreciated upon closer inspection, Kodosky's col. 18, line 35 - col. 20, line 2 and Kodosky's Figures 23A-32 teach using the graphical IDE, i.e. the graphical user interface, to create new graphical programs, i.e. block diagrams, or to open or retrieve existing graphical programs from non-volatile memory. It can be further appreciated that "To remedy shortcomings of the conventional data flow, which makes it substantially unusable as a programming language, the present invention includes graphical operators" (Kodosky col. 20, lines 21-24) (*emphasis added*). Kodosky further teaches using the graphical user interface or graphical IDE to include and use

graphical operators in graphical programs (i.e. block diagrams) in col. 20, line 28 - col. 42, line 7 and Figures 33-99.

In contrast, Applicant's invention as recited in claim 1 includes ". . .creating a graphical user interface for the graphical program in response to user input. . .(*emphasis added*)" Kodosky nowhere teaches or discloses this feature.

Applicant respectfully further submits Kodosky's Figure 25 illustrates user interaction with a graphical user interface (GUI) of a graphical development environment, specifically, to remove (or add) graphical program nodes to, or otherwise create or edit, a graphical program. In other words, Kodosky's Figure 25 (and related text) describes user interaction with the development environment's GUI to create a graphical program. Nowhere does Kodosky's Figure 25 show configuring an event structure node to receive and respond to user interface events during execution of the graphical program. Applicant respectfully submits that the Examiner has overlooked distinguishing features between the functionality of the development environment and that of the graphical program created via the environment.

For example, the Office Action asserts "configuring the event structure node to receive and respond to one or more user interface events during execution of the graphical program (the user can interact with the block diagram by utilizing icons to build the block diagram; furthermore, as an example, the user interface event "CLEAR", shown in Figure 25, can be used to remove certain wires from the block diagram). . ." Applicant respectfully submits that Examiner is referring to the graphical IDE in Kodosky. The "event 'CLEAR'" is handled by the graphical IDE in Kodosky and not a graphical program being constructed, displayed, or executed within the graphical IDE.

Thus, Applicant respectfully submits that Kodosky does not teach or suggest ". . .configuring the event structure node to receive and respond to one or more user interface events during execution of the graphical program" as recited in Applicant's claim 1. Kodosky nowhere teaches or suggests this feature.

Thus, Applicant respectfully submits that claim 1 is patentably distinguished over Kodosky. Accordingly, Applicant respectfully submits that, at least for one or more reasons presented, claim 1 and those dependent therefrom are allowable.

Claims 19, 23, and 32 include limitations similar to claim 1, and so the arguments presented above apply with equal force to these claims, as well. Applicant respectfully submits that for at least one or more reasons presented, claims 19, 23, and 32, and those claims, respectively, dependent thereon are patentably distinguished over Kodosky and are allowable.

Applicant respectfully requests removal of the §102 rejections.

§103 Rejections

Claims 8-9 were rejected under 35 U.S.C. 103(a) as being unpatentable over Kodosky and Zizzo (U.S. Pat. No. 6,578,174). This rejection is respectfully traversed.

Applicant respectfully submits that there is no teaching, suggestion, or motivation to combine Kodosky and Zizzo in either of the references or in the prior art. As held by the U.S. Court of Appeals for the Federal Circuit in *Ecolochem Inc. v. Southern California Edison Co.*, an obviousness claim that lacks evidence of a suggestion or motivation for one of skill in the art to combine prior art references to produce the claimed invention is defective as hindsight analysis. Furthermore, Applicant respectfully submits that it is nonobvious to combine Kodosky and Zizzo, and that even were Kodosky and Zizzo combinable, which Applicant argues they are not, the resultant combination would still not produce Applicant's invention as represented by claims 8-9.

The Office Action cites various of the dependent claims as being rejected under 35 U.S.C. 103. Various of the independent claims have been amended to overcome rejections under 35 U.S.C. 102, and Applicant submits that the amended independent claims are nonobvious and are allowable, as well. Applicant respectfully submits: "If an independent claim is nonobvious under 35 U.S.C. 103, then any claim depending

therefrom is nonobvious. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)" as stated in the MPEP §2143.03. Accordingly, Applicant respectfully submits that claims 1-66 are allowable.

Applicant also respectfully submits that numerous ones of the dependent claims recited further distinctions over the cited art. However, since the independent claims have been shown to be patentably distinct, a further discussion of the dependent claims is not necessary at this time.

Applicant respectfully requests removal of the §103 rejections.

CONCLUSION

In light of the foregoing amendments and remarks, Applicant submits the application is now in condition for allowance, and an early notice to that effect is requested.

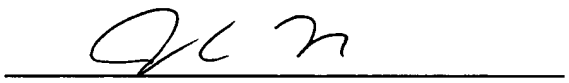
If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 50-1505/5150-58700/JCH.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

☒ Information Disclosure Statement

Respectfully submitted,



Jeffrey C. Hood
Reg. No. 35,198
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: 8/30/2004 JCH/MSW/IMF